Real-life Curriculum-based Timetabling

Tomáš Müller · Hana Rudová

June 2012

Abstract This paper presents an innovative approach to curriculum-based timetabling. Curricula are defined by a rich model that includes optional courses and course groups among which students are expected to take a subset of courses. Transformation of the curriculum model into the enrollment model is proposed and a local search algorithm generating corresponding enrollments is introduced. This enables curriculum-based timetabling in any existing enrollment-based course timetabling solver. The approach was implemented in a well established enrollment-based course timetabling system UniTime. The system has been successfully applied in practice at the Faculty of Education at Masaryk University for about 7,500 students and 260 curricula. Experimental results related with this problem are demonstrated for two semesters.

Keywords Course timetabling · Curriculum-based timetabling · Local search · UniTime

1 Introduction

Curriculum-based timetabling belongs to the class of university course timetabling problems (Burke and Petrovic, 2002; Lewis, 2008). Much research has been done in the area of curriculum-based timetabling (Di Gaspero et al, 2007; Bonutti et al, 2012), typically using a base curriculum model. In this

T. Müller
Space Management and Academic Scheduling, Purdue University
400 Centennial Mall Drive, West Lafayette, USA
E-mail: muller@unitime.org

H. Rudová
Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic
E-mail: hanka@fi.muni.cz

model, besides the usual classes, instructors and rooms tied together by various constraints (e.g., a room must be of large enough size, or an instructor can only teach one class at a time), there is a set of curricula defined. Each curriculum contains a list of courses that are to be attended by the same students (students of the curriculum). This is usually backed up by a hard constraint ensuring that classes of the same curriculum cannot overlap in time.

In the real world (McCollum, 2007), on the other hand, students are usually not required to attend all courses of a curriculum. Besides compulsory courses (courses that students must, or at least are expected, to take), there are elective courses (usually forming *groups*, where students are expected to take n of m courses) and optional courses that students may or may not take. Moreover, for some courses, students may decide during which semester they will take them. Typically, compulsory and elective courses cannot overlap in time, except there may be some overlaps of elective courses that are of the same group. For instance, if students are to take one of the given three courses, these three courses can be timetabled during the same time. Optional courses are usually only required to be at times that are not blocked by some other compulsory or elective course of the same curriculum. It is important to note here that each course may be present in multiple curricula, and it may be required for some curricula and only optional for another.

The whole problem is usually made even more complicated by the fact that courses tend to have multiple *course sections* (Hertz, 1991; Rudová et al, 2011). Courses with many students are usually split into several seminar groups and/or lectures. Furthermore, a course can be offered in various *configurations* (e.g., a lecture only, a lecture and a lab), with multiple lectures and labs available and some restrictions on what combinations of lecture and lab students are allowed to take. There can also be some mapping between curricula and specific classes of a course (e.g., the first lecture of a course may be reserved for students of an engineering major), but often there is none. Even simple course sectioning into several seminars leads to the inability to map curricula onto pairs of classes with no overlap in time. This is a very important aspect of the problem which must lead to more complex models and solutions.

1.1 Our Work

In our work, we rely on the UniTime[1] university timetabling system, containing an enrollment-based course timetabling solver which already deals with course sections and configurations of courses. In UniTime, students are assigned to classes based on student course demands (e.g., taken from pre-enrollment or from a previous semester) in a way that tries to keep students with similar courses together (Müller and Murray, 2010). A *class* is understood to be a part of the course which needs a time and a room assignment (e.g., each of the seminar groups or a lecture is a class). The course timetabling

[1] `http://www.unitime.org`

process looks for a proper assignment of times and classrooms to classes with the goal to optimize a set of criteria. It aims to minimize the number of student conflicts, i.e., cases when students are not able to attend classes due to their overlap in time or due to their placement of one right after the other in rooms that are too distant. Student conflicts can be decreased by assignment of a proper time as well as by swapping students between alternative classes or configurations of a course. Other important criteria include time and room preferences and restrictions for assignment of particular classes as well as distribution preferences specifying relations among several classes. A detail description of UniTime features and algorithms as well as its application to the Purdue University timetabling problem can be found at (Rudová et al, 2011).

This paper introduces a curriculum model which is applicable to solve real-life curriculum-based timetabling problems at large universities such as Masaryk University (Czech Republic) or Purdue University (USA). We also propose a transformation of this curriculum model into an enrollment model where each course is associated with a set of enrolled students (Lewis et al, 2007). A hybrid model of combining curricula with historic student enrollment data is also possible and briefly discussed in this paper. We describe a local search algorithm (Hoos and Stützle, 2005) which allows us to generate student course demands for the enrollment timetabling problem which respect characteristics of curricula. Given the curriculum model, transformation into the enrollment model, and the algorithm for generating student enrollments, it is possible to enable curriculum-based timetabling in *any* existing enrollment-based timetabling solver.

The proposed curriculum model and algorithm is implemented in UniTime and an application of this approach is presented on real-life curriculum-based timetabling problems from the Faculty of Education at Masaryk University. Timetables generated by UniTime have been used at the college in practice since Fall 2011. Here we have about 260 different curricula for about 7,500 students. This corresponds to timetabling of present, combined and lifelong forms of study. Overall computational results are presented on problems from two semesters, Fall 2011 and Spring 2012. We present results of the curricula to enrollments transformation for each of the three different forms of study, each representing a curriculum model with different characteristics. The number of curricula is very high since most of the programs in the college combine two different majors in order to educate teachers in two different subject areas (e.g., Math and Physics or Physics and Chemistry). A curriculum is defined for each allowed combination. This makes the overall timetabling very complex since there is a wide range of combinations with largely varying numbers of students in them. In particular, there are many combinations such as Math and Music, with only a few students, whose curriculum must still be respected. The initial requirement actually was to create timetables for compulsory and elective courses of all curricula with almost no student conflicts. In the paper, we demonstrate that timetabling of about 1,500 classes was possible with only about 100 student conflicts, i.e., 99.8 % of student course demands in curricula were satisfied for compulsory and elective courses. This was accompanied by consideration of a set of other problem characteristics including time and room preferences on individual classes as well as various relations among classes. During the timetabling process, the solution generated by fully automated methods was also interactively adjusted to reflect additional needs of the faculty and the students. The main results for the initial automated timetables, as well as for the final interactively corrected timetables that were used at the Faculty of Education in practice, are presented for both semesters.

In the following section we propose a curriculum model and the next chapter specifies a possible mapping between curriculum and enrollment models as a transformation of curricula to student course demands. Consequently we describe a local search algorithm that computes student course demands. Finally we provide experimental results from the application of the resultant system at the Faculty of Education.

2 Curriculum Model

We propose a curriculum model that is able to tackle an advanced set of real-life characteristics of timetabling problems. A curriculum, usually tied to students by their academic area (program of study) and one or more majors (further specializations) is split between different semesters. The tuple specifying curriculum and semester is called *classification*. A number of students is associated with each classification. This number may be known or may be estimated from previous semesters using various student projections. In addition to the number of students, each curriculum has an associated set of courses defined for each semester. Each course has a percentage which evaluates to the number of students that are expected to attend the course out of all those in the classification. These course percentages may also be replaced by the number of students from the classification expected to attend the course.

To model the relations between courses in a curriculum, various groups are defined. Each group contains a subset of courses in the curriculum. It is possible to create two types of groups. A *conflicting group* expresses that students are expected to take all courses in the group. A *non-conflicting group* indicates that students are expected to attend just one course in the group. During timetabling this means that courses in a non-conflicting group may overlap in time. Courses in a conflicting group must be timetabled so that all students in one course are able to attend all other courses in the conflicting group.

An illustrative example of a curriculum is presented in Figure 1. A bachelors degree in the curriculum is offered over three years. Students are required to take courses A and B during their first year of study, C and D during their second year of study, and E and F during their last year of study. They are also expected to take course G either the first or the second year (though 80 % of students usually take the course during the second year). Similarly, they can take course H during their second or third year. During the first

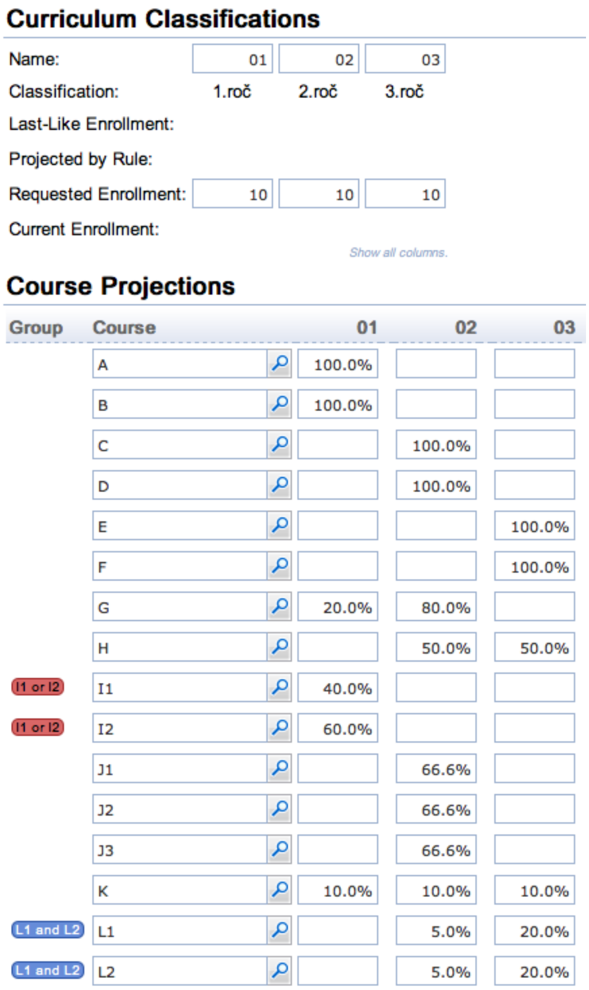


Fig. 1 Example of a curriculum prepared in UniTime timetabling system.

year, they should also take I1 or I2 (note that these two courses are put into a non-conflicting group "I1 or I2"). During their second year, they should also take two of courses J1, J2, J3 (this is solely modeled by the curriculum course percentages). There are also optional courses L1 and L2, which are either not taken at all or are taken together (this is modeled by the conflicting group "L1 and L2").

It is also possible for a course to be in multiple groups, as demonstrated in Figure 2. Due to the transitive closure relationship between groups discussed in Section 2.1, this allows modeling cases where a student needs to take a certain pair of courses (M1 together with M2 or N1 together with N2 or O1 together with O2) or other more complex cases.

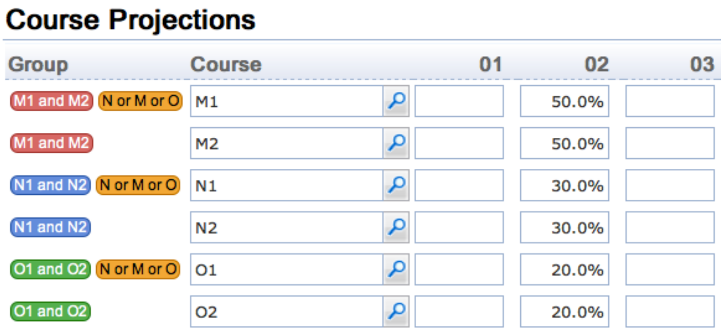


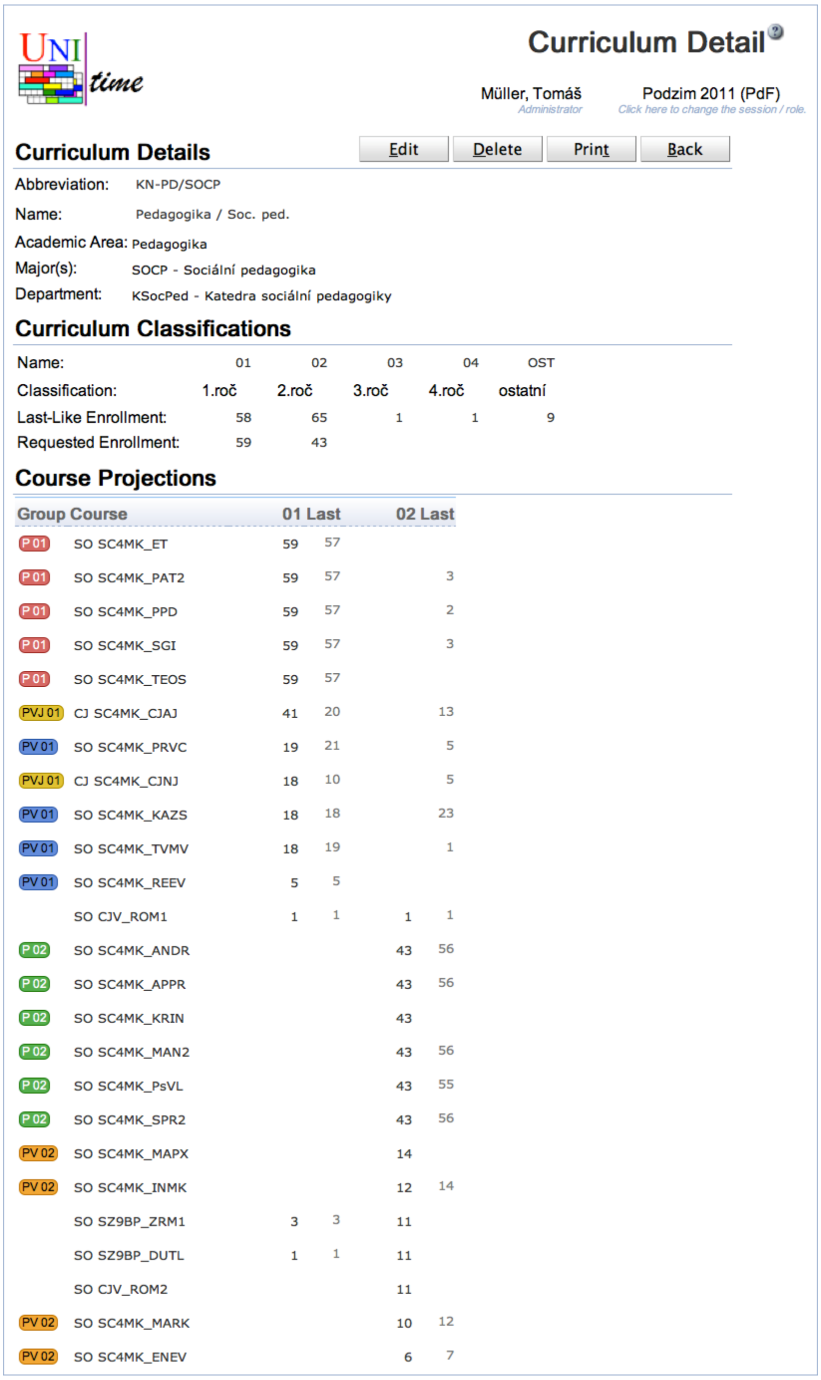
Fig. 2 Example with courses in multiple groups.

Please note that a curriculum may not contain all courses that a student in the curriculum may take during his/her study, but only courses that are offered in the term that is to be timetabled. For instance, if we are creating the Spring 2012 timetable, only courses that are offered in Spring 2012 will be present, but there will still be students associated with a curriculum that are in different years or semesters of their study. Also, if there has been a curriculum change starting in Spring 2011, students in their third year will take courses from the old curriculum whereas students in their first and the second years will need to follow the new curriculum.

A typical curriculum for the combined form of masters study taking two years is shown in Figure 3. In the *Course Projections* table, the columns titled *01* and *02* contain the expected enrollments for each course and semester. The columns titled *Last* are optionally displayed columns which indicate the number of students enrolled in the course during the last-like semester (Fall 2010). Courses in the conflicting groups P 01 and P 02 represent compulsory courses in the first and the second year respectively. Courses in the non-conflicting groups PV 01, PVJ 01, PV 02 represent elective courses. Students in this curriculum are expected to take two electives in the first year (one from each group PV 01 and PVJ 01) and only one elective in the second year (group PV 02). Optional courses do not belong to any group (column *Group* in the *Course Projections* table is empty). Note that the groups P 01 and P 02 are not necessary as all the courses in these two groups expect attendance by all the first year or second year students respectively.

2.1 Formal Model

More formally, there is a set of courses $c \in C^a$ for each curriculum $a \in A$. The set of all courses in the whole timetabling problem corresponds to $\bigcup_{a \in A} C^a$. Since a course may appear in more than one curriculum, sets C^a and C^b for curricula $a, b \in C$ may possibly have a non-empty intersection. A curriculum a is associated with student counters $x_1^a, x_2^a, \ldots x_n^a$ where n is the number of semesters and x_i^a is the number of students in curriculum a and semester i.



Curriculum Detail

Müller, Tomáš
Administrator

Podzim 2011 (PdF)
Click here to change the session / role.

Curriculum Details

Edit Delete Print Back

Abbreviation: KN-PD/SOCP
Name: Pedagogika / Soc. ped.
Academic Area: Pedagogika
Major(s): SOCP - Sociální pedagogika
Department: KSocPed - Katedra sociální pedagogiky

Curriculum Classifications

Name:	01	02	03	04	OST
Classification:	1.roč	2.roč	3.roč	4.roč	ostatní
Last-Like Enrollment:	58	65	1	1	9
Requested Enrollment:	59	43			

Course Projections

Group	Course	01	Last	02	Last
P 01	SO SC4MK_ET	59	57		
P 01	SO SC4MK_PAT2	59	57		3
P 01	SO SC4MK_PPD	59	57		2
P 01	SO SC4MK_SGI	59	57		3
P 01	SO SC4MK_TEOS	59	57		
PVJ 01	CJ SC4MK_CJAJ	41	20		13
PV 01	SO SC4MK_PRVC	19	21		5
PVJ 01	CJ SC4MK_CJNJ	18	10		5
PV 01	SO SC4MK_KAZS	18	18		23
PV 01	SO SC4MK_TVMV	18	19		1
PV 01	SO SC4MK_REEV	5	5		
	SO CJV_ROM1	1	1	1	1
P 02	SO SC4MK_ANDR			43	56
P 02	SO SC4MK_APPR			43	56
P 02	SO SC4MK_KRIN			43	
P 02	SO SC4MK_MAN2			43	56
P 02	SO SC4MK_PsVL			43	55
P 02	SO SC4MK_SPR2			43	56
PV 02	SO SC4MK_MAPX			14	
PV 02	SO SC4MK_INMK			12	14
	SO SZ9BP_ZRM1	3	3	11	
	SO SZ9BP_DUTL	1	1	11	
	SO CJV_ROM2			11	
PV 02	SO SC4MK_MARK			10	12
PV 02	SO SC4MK_ENEV			6	7

Fig. 3 Example of curriculum from Fall 2011 at the Faculty of Education.

Note that the tuple (a, i) defines a classification. To simplify formulations, we define the set AI as a set of all classifications (a, i).

Furthermore, there is a matrix with values $e^a_{c,i}$ between 0 and 1. For each course c and semester i, the value $e^a_{c,i}$ defines the proportion of the number of students x^a_i expected to attend the course c. Each course $c \in C^a$ is expected to be attended by $e^a_{c,i} x^a_i$ students from classification (a, i). Finally, there are groups of courses $G^a_1 \subseteq C^a, \ldots, G^a_k \subseteq C^a$ representing conflicting groups and $H^a_1 \subseteq C^a, \ldots, H^a_l \subseteq C^a$ representing non-conflicting groups in the curriculum.

It is also important to mention that for each pair of courses $c, d \in C^a$, we expect the following proportion of the number of students in classification (a, i) represented by values from interval $\langle 0, 1\rangle$.

$$t^a_{c,d,i} = \begin{cases} 0 & \exists j : c, d \in H^a_j, \\ 1 & \exists j : c, d \in G^a_j, \\ e^a_{c,i} e^a_{d,i} & \text{otherwise.} \end{cases}$$

We call this number the *target share* of a curriculum between the two courses.

For the above example with Figure 1, the target share between courses L1 and L2 is 1 (all students attending L1 are expected to attend L2 and vice versa), it is 0 between courses I1 and I2 (students are taking either I1 or I2, but not both), and it is 0.44 between J1 and J2 for the second year students (44 % of students are expected to take both J1 and J2).

If there are courses in multiple groups, a special graph needs to be considered. Here the nodes are represented by courses and the edges are defined by the existence of a (conflicting or non-conflicting) group between the two courses. More precisely, there is an edge between $c, d \in C^a$ if courses c and d are present in the same group. The target share $t^a_{c,d,i}$ is set to zero if there is a path $c = c_1, c_2, \ldots, c_{m-1}, c_m = d$ in the graph where at least one of the groups defining edges on the path is non-conflicting. The target share $t^a_{c,d,i}$ is set to one if the path has all the groups conflicting. Note that correct computation of all target share values necessitates computation of the transitive closure in the graph. For the above example with Figure 2, this means that we also expect no students to be between course M1 and N2, between M2 and N1, and between M2 and N2 (and similar for O1 and O2 courses).

3 Curriculum to Enrollment Model Transformation

We now show that the proposed curriculum model can be transformed into an enrollment model. In the enrollment model, each course has a set of students enrolled in it. Our goal is to find an assignment of students to courses in the enrollment model such that the curricula are respected. This means that student conflicts in the enrollment model should correspond with the number of broken requests given by curricula. For instance, two compulsory courses in a curriculum with 10 students timetabled at the same time corresponds to 10 student conflicts in the enrollment model.

First, consider the target share between two courses in a curriculum. We propose relating the target share with the number of students who must be able to attend both courses. This corresponds with the number of student conflicts in the enrollment model. Further, particular target shares specify the characteristics of an *ideal* enrollment model in which student conflicts fully correspond with the unmet curricular course requirements in the curriculum model. Having this in mind, we define an optimization criterion evaluating assignments of students to courses in the enrollment model. This criterion evaluates the affinity of this assignment to an ideal enrollment model. Generally, we can summarize distances from optimality based on a comparison of characteristics (target shares) of the ideal enrollment model with characteristics of the assignment. Certainly our goal is to find an assignment with the minimal distance. For instance, if there are two courses with target share of 15 students in a curriculum a and our assignment has only 14 students that are enrolled in both of these courses from a, then the contribution to the distance corresponds to 1.

3.1 Formal Transformation

In the curriculum model, there are x_i^a students for each classification (a, i). We summarize characteristics of the ideal enrollment model with respect to the curriculum model.

1. There are $e_{c,i}^a x_i^a$ students from the classification (a, i) enrolled in the course c.
2. The target share $t_{c,d,i}^a$ between courses c and d of the classification (a, i) specifies the number of students $t_{c,d,i}^a x_i^a$ enrolled in both courses c and d.

If we expect that each student is enrolled in only one curriculum[2], all curricula are independent and do not share any students. Also, it is clear that each student is enrolled in only one semester of the curriculum a. This means that there are different students in the two semesters i, j of curriculum a. This extends characteristics of the ideal enrollment model with respect to the curriculum model.

1. There is a total of $\sum_{(a,i)\in AI} e_{c,i}^a x_i^a$ students enrolled in the course c.
2. For each two courses $c, d \in C^a$, the total number of students enrolled in both courses corresponds to $\sum_{(a,i)\in AI} t_{c,d,i}^a x_i^a$.

Next we consider an assignment θ of students to courses defining an enrollment model which will be evaluated with respect to the ideal enrollment model. We also expect that each such student belongs to a curriculum $a \in A$ and its semester i and the number of students in curriculum a equals to x_i^a for the semester i. In the assignment θ, we denote the number of students x_i^a in classification (a, i) belonging to both courses $c, d \in C^a$ as an *actual share* $s_{c,d,i}^a$.

[2] This is typical for the vast majority of students in our case. Only students wanting to study two different topics would not satisfy this condition.

The assignment θ is evaluated by the *distance*

$$F(\theta) = \sum_{(a,i)\in AI} \sum_{c,d\in C, c\neq d} |t^a_{c,d,i} x^a_i - s^a_{c,d,i}| = \sum_{(a,i)\in AI} F(\theta, a, i) \ .$$

Clearly an ideal assignment ω (assignment with the ideal enrollment model) has a distance $F(\omega) = 0$. Since such an assignment may not necessarily exist, our goal is to find an assignment σ with the minimal distance $F(\sigma)$. We also defined $F(\theta, a, i)$ since students in each classification (a, i) are different from students in all other classifications. Following that, the assignment of students for each classification is independent of assignments in all other classifications and the following statement holds.

$$\min F(\theta) = \sum_{(a,i)\in AI} \min F(\theta, a, i) \qquad (1)$$

3.2 Using Historical Data

It is also possible to make adjustments to the target share matrix based on historical data, typically from the last-like semester (e.g., last-like semester for Spring 2012 is Spring 2011). For instance, if we know from past experience that students attending J1 are more likely to attend J2 than J3, we can use this information to adjust the values of the matrix to reflect this.

More formally, the target share is $t^a_{c,d,i} = p\, r^a_{c,d,i} + (1-p) e^a_{c,i} e^a_{d,i}$ where $r^a_{c,d,i}$ is the percentage of students from classification (a, i) that took both courses c and d in the last-like semester, and p a number between 0 and 1 defining how much we want to stick with the past. Note that this new target share only applies to pairs of courses for which we have historical data. In other words, if either course c or d is newly offered, the target share is only defined by $e^a_{c,i} e^a_{d,i}$ matrices and the groups in the previous chapter.

Given this, we can take assignment θ of students in the last-like semester, compute its distance $F(\theta)$ and consider it as an initial assignment when looking for an (sub-)optimal solution to the timetabling problem using the curriculum model.

4 Construction of Enrollments

We specified how students should be assigned to courses to respect the curriculum model by minimization of the distance $F(\theta)$. In this section, we describe a local search algorithm which allows computation of a "reasonable" sub-optimal assignment θ of students to courses with respect to $F(\theta)$. As discussed in the second paragraph of Section 3.2 and demonstrated in Equation 1, we expect different students for each classification (a, i). This means that the assignment θ can be computed per partes, i.e., the local search algorithm is applied to each classification separately.

First, it is important to discuss computation of the target share between two courses $c, d \in C^a$. We will concentrate on computation of the number of students $\bar{t}^a_{c,d,i}$ that are expected to be assigned to both courses c and d. Corresponding to Section 2.1, this is counted based on $t^a_{c,d,i} x^a_i$. However, it is also bounded by the number of students in courses c and d, respectively. To account for this, the number of students in the course c for classification (a, i) is denoted $x^a_{c,i} = \text{round}(e^a_{c,i} x^a_i)$. For instance, if there are 20 students in the given semester of a curriculum, and courses c and d are expected to be attended by 10 and 15 students respectively, the share of the two courses must be between 5 and 10.

$$\bar{t}^a_{c,d,i} = \begin{cases} \max(0, x^a_{c,i} + x^a_{d,i} - x^a_i) & \exists j : c, d \in H^a_j, \\ \min(x^a_{c,i}, x^a_{d,i}) & \exists j : c, d \in G^a_j, \\ \max\Big(\min\Big(\text{round}(e^a_{c,i} e^a_{d,i} x^a_i), x^a_{c,i}, x^a_{d,i}\Big), x^a_{c,i} + x^a_{d,i} - x^a_i\Big) & \text{otherwise.} \end{cases}$$

The overall distance $F(\theta, a, i)$ for classification (a, i) is counted incrementally for each course as the difference $\Delta F(\theta, a, i, c, z_{\text{new}}, \perp)$ between the distance before and after assignment of a student z_{new} into a course c. It also allows for a swap of a course between two students z_{new} and z_{old} (denoted by $\Delta F(\theta, a, i, c, z_{\text{new}}, z_{\text{old}})$). Pseudo-code of this function is presented in Figure 4.

```
function ΔF(θ, a, i, c, z_new, z_old)
  f = 0
  for d ∈ C^a such that d ≠ c
      (for each course other than c, f is increased by the difference
      between target share and actual share before and after the change)
      t := t̄^a_{c,d,i}
      s := s^a_{c,d,i}
      f := f − |t − s|
      if z_new ∈ students(d) then s := s + 1
      if (z_old ≠⊥ and z_old ∈ students(d)) then s := s − 1
      f := f + |t − s|
  return f
```

Fig. 4 Pseudo-code of function $\Delta F(\theta, a, i, c, z_{\text{new}}, z_{\text{old}})$.

The search for assignment of students to courses for the given classification (a, i) is processed in two phases as can be seen in Figure 5. In the first phase (lines 3-7) a simple *construction* heuristic is used. In each iteration, a single student is assigned to a particular course until each course has the desired number of students. Courses are ordered dynamically by the number of remaining spaces (if there are two or more courses with the same number of remaining spaces, one is selected randomly — see line 4). For a selected course, all students that are not yet assigned to it are checked, and one of the students that has the best impact on the overall distance is selected randomly (line 5).

In the second phase (lines 8-18) a *great deluge* approach (Dueck, 1993) is used. The initial bound UB is set to 1.25 of the initial solution's distance f

1: **procedure** SEARCH(a, i, α)
2: $f := 0$
(construction phase)
3: **while** $\exists u \in C^a$ **such that** $\|students(u)\| < x^a_{c,i}$
4: $c :=$ RANDOM($d \in C^a$ **such that** MAXIMAL$(x^a_{d,i} - \|students(d)\|)$)
5: $z :=$ RANDOM($v \notin students(c)$ **such that** MINIMAL$(\Delta F(\theta, a, i, c, v, \perp))$)
6: $f := f + \Delta F(\theta, a, i, c, z, \perp)$
7: $students(c) := students(c) \cup \{z\}$
(great deluge phase)
8: $UB = 1.25 * f$ (upper bound)
9: **while** ($UB \geq 0.75 * f$ **and** $f > 0$)
10: $c :=$ RANDOM($d \in C^a$ **such that** $\|students(d)\| < x^a_{d,i}$ **and** $x^a_{d,i} > 0$)
11: $z_{\text{old}} :=$ RANDOM$(v \in students(c))$
12: $z_{\text{new}} :=$ RANDOM$(v \notin students(c))$
13: $\Delta f := \Delta F(\theta, a, i, c, z_{\text{new}}, z_{\text{old}})$
14: **if** ($\Delta f \leq 0$ **or** $f + \Delta f \leq UB$) **then**
15: $f := f + \Delta f$
16: $students(c) := students(c) \setminus \{z_{\text{old}}\}$
17: $students(c) := students(c) \cup \{z_{\text{new}}\}$
18: $UB := UB * (1 - \alpha)$

Fig. 5 Pseudo-code of the algorithm computing enrollments for classification (a, i).

(line 8) and it is decreased by the coefficient α (typically 0.0001 %) in each iteration (line 18). The search is stopped when a solution with zero distance is found or when the bound reaches 0.75 of the solution's distance (line 9). In each iteration, a possible change of a single student in a course is generated (lines 10-12) and accepted if the resultant solution's distance does not exceed the bound (lines 14-17). Changes that do not increase the distance are also accepted. A course, a student that is removed from this course and a student that is assigned to this course are selected randomly (lines 10, 11, 12, respectively).

When an assignment θ with zero distance $F(\theta, a, i)$ is found during the first phase, the second phase is not executed. When historical data are available and we want to take them into account, the first phase starts with last-like semester enrollments (see Section 3.2).

5 Experimental Results

The following experiments are based on Fall 2011 and Spring 2012 data from the Faculty of Education at Masaryk University. Fall 2011 is the first semester for which UniTime was used to build the course timetable for the college. Experiments in Section 5.1 were computed on an Apple MacBook Pro with a 3.06 GHz Intel Core 2 Duo processor and 8 GB RAM, running Mac OS X 10.7.3 and Java 1.6.0. Results in Section 5.2 are presented from the installation for the Faculty of Education and the Faculty of Arts (Rudová and Müller, 2011) running on a virtual machine hosted on a machine with two Intel X5560 processors and 96 GB RAM (8 GB dedicated to the UniTime virtual machine).

5.1 Curriculum to Enrollment Transformation

Table 1 shows results from the curriculum to enrollment solver. All of the curricula are split into three groups based on the student's form of study. Results are presented for all curricula together as well as for particular sets. For each set, the number of curricula, classifications and students is specified. We also present the number of students per classification and the number of courses per classification. For these data sets, the average distance $F(\theta, a, i)$, the average distance achieved after the construction phase of the search algorithm (see Figure 5), and the average computational time is presented for 10 independent runs.

We can see that curricula of the present form of study introduce the largest data set with the highest computed distance. Curricula of the combined form of study can be transformed into a better enrollment model (the distance is lower) and it takes a longer time. Curricula of the lifelong form of study introduce the smallest data set and are easiest to transform. To understand the achieved quality of the solution we note that the distances 7.05 and 6.66 for the present form of study correspond to 0.60 % and 0.69 % of the worst possible distance, respectively.

Certainly we tried to find the best possible distance $F(\theta, a, i)$ in a reasonable time. We ran the set of experiments with varying size of α (see line 18 of Figure 5 and description of the algorithm) influencing progress of the search

Fall 2011	All together	Present (P)	Combined (K)	Lifelong (C)
Spring 2012				
Curricula	265	210	28	27
	258	202	25	31
Classifications	574	470	56	56
	543	442	53	48
Students	7,569	4,301	2,562	706
	6,803	3,852	2,362	589
Students	13.19	9.15	45.75	14.71
per classif.	12.53	8.71	44.57	12.27
Courses	30.61	34.63	18.32	5.67
per classif.	27.44	31.06	15.62	7.21
$F(\theta, a, i)$	7.05 ± 0.01	8.24 ± 0.01	3.14 ± 0.03	0.00 ± 0.00
	6.66 ± 0.01	8.04 ± 0.01	1.02 ± 0.03	0.13 ± 0.00
$F(\theta, a, i)$	11.97 ± 0.13	13.25 ± 0.13	11.53 ± 0.14	0.04 ± 0.06
after 1. phase	10.87 ± 0.11	11.99 ± 0.12	11.03 ± 0.12	0.31 ± 0.06
CPU time [s]	3.36 ± 0.06	3.08 ± 0.05	8.58 ± 0.12	0.01 ± 0.00
	3.53 ± 0.07	2.88 ± 0.06	12.14 ± 0.16	0.02 ± 0.03

Table 1 Computing enrollments for two semesters.

algorithm. The smaller α is the slower the upper bound decreases and the great deluge algorithm has more time for optimization. Results of this experiment are available in Figure 6. Here we can see that the algorithm is able to

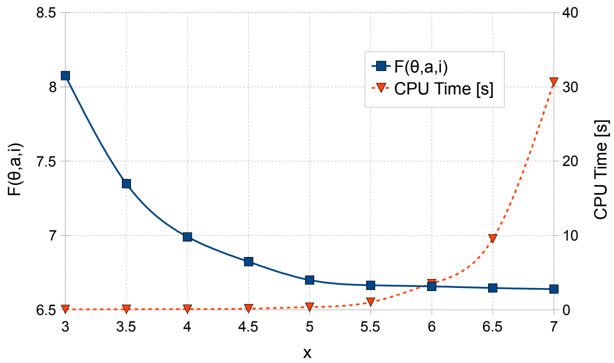


Fig. 6 Graph depicting dependency of the distance $F(\theta, a, i)$ and the computational time on the value $\alpha = 10^{-x}$ for the problem with all curricula and semester Spring 2012.

improve the distance with a reasonable demands on computational time up to the value $\alpha = 0.0001\,\%$ (corresponding to $x = 6$ in the graph). Decreasing this value further does not achieve significant improvements in the distance in a reasonable computational time.

5.2 Course Timetabling

Table 2 contains results from the course timetabling (courses from the present form of study only) at the Faculty of Education. For each semester, there are results from the automated solver as well as the published solution, after a few interactive changes were made. Note that published solutions were used in practice. The table shows the overall number of courses and the number of compulsory and elective courses (in parenthesis). Similarly, the number of classes and the number of student enrollments in each problem are presented. The second part of the table presents the main characteristics of the computed solution. The most important factor of the problem was the number of student conflicts among compulsory and elective courses. As we can see, we have only 112 and 96 conflicts for 1,575 and 1,408 timetabled classes, respectively. This is certainly a very strong result given all of the complexities and the size of both problems (recall from Table 1 that we have 210 and 202 curricula for 4,301 and 3,852 students, respectively). The number of student conflicts among all courses is slightly higher, it is mostly due to overlaps between optional and compulsory or elective courses. These numbers, together with results for time, room, and distribution preferences, correspond with priorities of the school and the importance of particular criteria. The better results for Spring semester were achieved due to a smaller number of classes timetabled into the same

	Fall 2011 automated	Fall 2011 published	Spring 2012 automated	Spring 2012 published
Courses (comp. & elect.)	1,225 (1,156)		900 (870)	
Classes (comp. & elect.)	1,831 (1,575)		1,665 (1,408)	
Enrollments (comp. & elect.)	57,861 (52,396)		45,786 (45,400)	
Student conflicts	418 (0.63 %)	456 (0.69 %)	477 (1.02 %)	417 (0.89 %)
among comp. & elect.	112 (0.17 %)	140 (0.21 %)	96 (0.20 %)	93 (0.20 %)
Time preferences	89.27 %	89.93 %	94.88 %	95.32 %
Room preferences	78.03 %	79.92 %	85.15 %	86.50 %
Distribution preferences	84.50 %	80.41 %	90.49 %	90.49 %
Interactive changes		355		275
of time		183		105
of room		300		218

Table 2 Results from timetabling at the Faculty of Education for two semesters.

amount of available classrooms. Finally, the number of interactive changes with the initial solution is demonstrated. These low numbers show that it is not necessary to adjust solutions much. Still, it is important to allow some adjustments to the solutions to make them more acceptable for the school.

6 Conclusion

We presented a new approach to curriculum-based timetabling and applied it to solve large-scale problems at the Faculty of Education where it is implemented in the UniTime system and used for timetabling since Fall 2011. Automated timetabling simplified the process for the college where about 40 schedule deputies cooperated on creating timetables manually until Spring 2011. Presently, they provide only inputs, such as desirable assignments of times and rooms to classes, and the timetables are constructed by UniTime. It is also important to mention the existence of Information System[3] at Masaryk University where curriculum data are maintained and can be used for timetabling directly. Resultant timetables are also available here[4].

Further generalization of the approach involves inclusion of students in more than one curriculum. This is fully compatible with the proposed search algorithm where students can be used in different curricula and additional courses will be generated for them from each curriculum. However, this approach is not yet implemented. Also, some reformulation of the curriculum model related to the proposed distance function is necessary. Our intent to include this functionality lies in easier management of existing curricula. Having many curricula composed of two different majors (examples are Math and

[3] `http://is.muni.cz`
[4] `http://is.muni.cz/rozvrh/?fakulta=1441&lang=en`

Physics or Physics and Chemistry as discussed before), it would be easier to maintain the smaller set of majors and an additional set of acceptable combinations of majors defining curricula.

Acknowledgements This work is supported by the Grant Agency of Czech Republic under the contract P202/12/0306. The access to the MetaCentrum computing facilities provided under the program "Projects of Large Infrastructure for Research, Development, and Innovations" LM2010005 funded by the Ministry of Education, Youth, and Sports of the Czech Republic is highly appreciated.

References

Bonutti A, De Cesco F, Di Gaspero L, Schaerf A (2012) Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. Annals of Operations Research To appear

Burke EK, Petrovic S (2002) Recent research directions in automated timetabling. European Journal of Operational Research 140:266–280

Di Gaspero L, McCollum B, Schaerf A (2007) The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Tech. Rep. QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0, University, Belfast, United Kingdom

Dueck G (1993) New optimization heuristics: The great deluge algorithm and the record-to record travel. Journal of Computational Physics 104:86–92

Hertz A (1991) Tabu search for large scale timetabling problems. European Journal of Operational Research 54(1):39–47

Hoos HH, Stützle T (2005) Stochastic Local Search Foundations and Applications. Elsevier

Lewis R (2008) A survey of metaheuristic-based techniques for university timetabling problems. OR Spectrum 30(1):167–190

Lewis R, Paechter B, McCollum B (2007) Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Cardiff Working Papers in Accounting and Finance A2007-3, Cardiff Business School, Cardiff University

McCollum B (2007) A perspective on bridging the gap between theory and practice in university timetabling. In: Burke E, Rudová H (eds) Practice and Theory of Automated Timetabling VI, Springer-Verlag LNCS 3867, pp 3–23

Müller T, Murray K (2010) Comprehensive approach to student sectioning. Annals of Operations Research 181:249–269

Rudová H, Müller T (2011) Rapid development of university course timetables. In: Proceedings of the 5th Multidisciplinary International Scheduling Conference – MISTA 2011, pp 649–652

Rudová H, Müller T, Murray K (2011) Complex university course timetabling. Journal of Scheduling 14(2):187–207